

COTS software in science operations, is it worth it?

William O'Mullane¹ and Nana Bach² and Jose Hernandez¹ and Alexander Hutton² and Rosario Messineo³,

¹ *European Space Astronomy Centre, P.O. Box 78, 28691 Villanueva de la Cañada, Spain*

² *Aurora for ESA at ESAC*

³ *Altec Turin, Italy*

1. Introduction

The Gaia astrometric satellite is now in operations for more than two years. The first data release (Gaia Collaboration et al. 2016) was highly successful. Behind the operations there is a lot of software and collaboration. The Data Processing Ground Segment is jointly operated by ESAC and the Gaia DPAC. This is comprised of the Science Operations Centre (SOC) operated by ESA and a set of Data Processing Centres (DPCs). The SOC is also a DPC (DPCE for daily and astrometric processing), SOC also acts as the interface between the Mission Operations Centre (MOC) and the DPAC (See Figure 1). Gaia produces an impressive volume of raw data with about 50GB of uncompressed science data per day, yielding at mission completion a telemetry data volume of roughly 500TB. Transforming the data into scientifically meaningful quantities is the task of the Data Processing Analysis Consortium (DPAC). DPAC is comprised of a number of Coordination Units (CUs). Each one is responsible for a well-defined part of the Gaia data processing (Mignard et al. 2008). For each CU there is at least one Data Processing Center (DPC) with dedicated resources for the data processing of the CU.

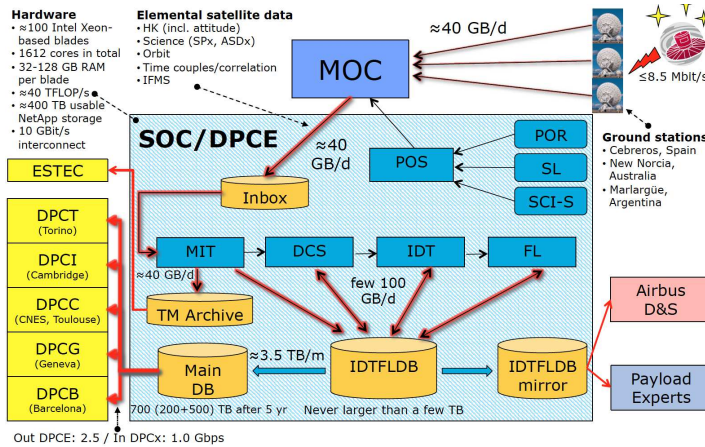


Figure 1. Gaia SOC/DPCE at ESAC

2. The Main Database

The MDB (Main Database) is the central repository of all the data produced by Gaia Scientific Processing and DPAC. The MDB Dictionary, running at DPCE, is used by all to define the Datamodel of the MDB. There are actually a bunch of software systems incorporated in the MDB:

- MDB Database
- MDB Schema Creator
- MDB Dictionary Tool
- MDB Extractor
- MDB Ingestor
- MDB Integrator
- MDB Explorer
- MDB Data Manager

There are documents listing MDB Requirements and they may be summarised as follows:

- **14 functional requirements**
- **1 performance requirement**
 - MDB Schema Creator: 1 functional requirement
 - MDB Dictionary Tool: 24 functional requirements
 - MDB Extractor: 12 functional requirements, 1 performance requirement
 - MDB Ingestor: 10 functional requirements, 2 performance requirements
 - MDB Integrator: 9 functional requirements
 - MDB Explorer: 16 functional requirements
 - Organically added functionality
 - * MDB DataManager: no written requirements

3. Gaia Transfer System (GTS)

GTS (Gaia Transfer System) is the service that permits data exchange between the Data Processing Centres (DPCs). This is composed of Aspera (commercial) and the DTSTool (add on built within DPAC by Altec). Aspera is a COTS for fast data transfer, it provides the technical platform for the data transfers. We have found this an extremely useful product with very good support.

The DTSTool is responsible for building an interface between Aspera and the data processing software systems at each DPC.

Again we have documents containing GTS Requirements which may be summarised as:

- **13 functional requirements**
- **2 performance requirements**
- 29 other requirements:
 - Portability: 1
 - User Interface: 5
 - Execution and activation: 1
 - File naming convention: 3
 - Interface: 5
 - Safety and Security: 12
 - Performance: 3

4. Cost-effectiveness analysis - factors

For serious implementations, the real cost of software goes far beyond the license itself, and includes (see also et Al. (2017)):

- Time for implementation, evaluation, and integration, heavily influenced by:
 1. The quality of documentation
 2. The responsiveness of support staff
 3. The availability of code examples and other learning aides
- Resources consumed by the use of the software.
- Reliability of the software across all possible use-cases (often the most *expensive* aspect of cheaper software).
- Flexibility and adaptability of the software relative to the competition.

To achieve real cost-effectiveness we recommend seeking the following qualities in server software for high-demand workflows:

- Memory-allocation independent (application terminates after execution).
- Fully separable from other functions.
- Availability of well defined and stable API in order to decouple the COTS from SW customisation.
- Proven track-record of reliability in high-volume production environments.

4.1. MDB Cost

We have a reasonable account of effort booked to the MDB work package 2005-2016:

- Effort \approx 18 person years incl. testing and documentation/support.
- Depending on our cost model that is a cost of €2.8M to €4.5M Bear in mind this must include all consumables, office space, phones, travel etc.. It also contains management overhead not just an individuals salary.
- Lets call it €3M.

4.2. GTS Cost

For Aspera we have a mix of licenses and development:

- Aspera licenses and support to date €158K
- DTSTool, customisation, automation etc. .. effort \approx 3.5 person years incl. testing documentation.
- Effort cost (using a similar rate to MDB but actually probably lower) €560k
- Total cost \approx 720k

4.3. Cost effectiveness

There is No good way to do this! Constructive Cost Model (COCOMO) good *BEFORE* you build perhaps. Though similar in requirement the MDB feels much more complex than GTS if we use the functional requirements and say all were met that gives MDB a much higher complexity with 86 v 29 requirements.

- UNIT COST = Cost/Function
- MDB UC= $3M/86 = 34883$
- GTS UC= $720K/29 = 24827$

COST EFFECTIVENESS could be considered as the Ratio of in house to COTS software unit development cost. This would make COTS 1.4 times more cost effective. Of course we can only use COTS where appropriate, we could not find a tool or set of tools to do the job of the MDB.

5. Conclusion

To do this more effectively we should try harder to scope functional requirements in comparable way. The COTS items we can use in space science is fairly limited (DBMS, xfer other generic stuff). But we should choose carefully - that small development can cost a lot cumulatively over our very long projects.

Open source is ok but can also die out over our long projects. In Gaia we supported Apache Common Math - now ONLY we support it as the community seems to have disappeared. Java as a sort of open platform worked out ok for Gaia do far - who knows what will happen in the next decade. The Eclipse IDE , MANTIS (now we use Jira) are all good there are peripheral tools which work and save a lot.

References

- et Al., R. G. 2017, in ADASS XXVI, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD
- Gaia Collaboration, Brown, A. G. A., Vallenari, A., Prusti, T., de Bruijne, J., Mignard, F., Drimmel, R., & co-authors, . 2016, ArXiv e-prints. 1609.04172
- Mignard, F., Bailer-Jones, C., Bastian, U., Drimmel, R., Eyer, L., Katz, D., van Leeuwen, F., Luri, X., O'Mullane, W., Passot, X., Pourbaix, D., & Prusti, T. 2008, in A Giant Step: from Milli- to Micro-arcsecond Astrometry, edited by W. J. Jin, I. Platais, & M. A. C. Perryman, vol. 248 of IAU Symposium, 224. 0712.0889